
Algorithme approché pour un problème de partitionnement de maillage sous contrainte mémoire

Sébastien Morais^{*†2,1}, Eric Angel^{‡1}, Cédric Chevalier^{§2}, Franck Ledoux^{¶2}, Kim Thang Nguyen^{||1}, and Damien Regnault^{**1}

²Commissariat à l'énergie atomique et aux énergies alternatives (CEA) – CEA – France

¹Informatique, Biologie Intégrative et Systèmes Complexes (IBISC) – Université d'Evry-Val d'Essonne :
EA4526 – 23, Bd de France; 91034 - EVRY Cedex, France

Résumé

La taille et la complexité des simulations numériques rendent souvent nécessaire l'utilisation de calculateurs à mémoire distribuée. La répartition des données est alors cruciale : elle doit minimiser le coût de calcul tout en assurant que les données nécessaires à chaque processeur puissent être stockées localement en mémoire.

Nous considérons le cas de simulations, où les données physiques sont supportées par un maillage. Le calcul peut alors être réparti par maille, chacune ne nécessitant que la connaissance des données de la maille et de ses voisines. La répartition des calculs correspond alors à un partitionnement du maillage : chaque maille est attribuée à un unique processeur, les données des mailles voisines étant dupliquées si nécessaire (mailles fantômes).

Les approches usuelles de partitionnement ignorent généralement les mailles fantômes. Dans notre étude, nous nous intéressons au partitionnement de maillage sous contrainte mémoire en considérant la notion de voisinage. Nous l'étudions comme un problème d'affectation minimisant la somme des durées d'exécution sur tous les processeurs (supposés hétérogènes), où la mémoire consommée par chaque maille est unitaire.

Formellement, nous cherchons un partitionnement tel que la somme des durées d'exécution soit au plus C et tel que l'occupation mémoire pour chaque processeur soit au plus M . Ce problème s'exprime comme un programme linéaire en nombres entiers $P(C, M)$.

Afin de résoudre P , nous proposons un algorithme (en temps polynomial) approché avec garantie de performance. Celui-ci consiste en trois étapes: (1) Résoudre la relaxation linéaire $PL(C, M)$ de P ; (2) Créer un graphe biparti particulier à partir de la solution obtenue; (3) Rechercher un couplage maximum (calcul/processeur) de coût minimum.

*Intervenant

†Auteur correspondant: sebastien.morais@free.fr

‡Auteur correspondant: Eric.Angel@ibisc.univ-evry.fr

§Auteur correspondant: cedric.chevalier@cea.fr

¶Auteur correspondant: franck.ledoux@cea.fr

||Auteur correspondant: thang@ibisc.fr

**Auteur correspondant: damien.regnault@ibisc.fr

Nous avons prouvé que si une solution du programme linéaire $PL(C,M)$ existe, alors la méthode précédente permet de trouver une solution pour $P(C,(A+1)M)$ en temps polynomial, où A est le nombre maximum de voisins pour une maille. Détails sur l'algorithme : <https://www.ibisc.univ-evry.fr/~smorais/rapport.pdf> chapitre 3

Mots-Clés: Partitionnement, maillage, algorithme approché, programmation linéaire